



Amazon Web Services

Data Engineering Immersion Day

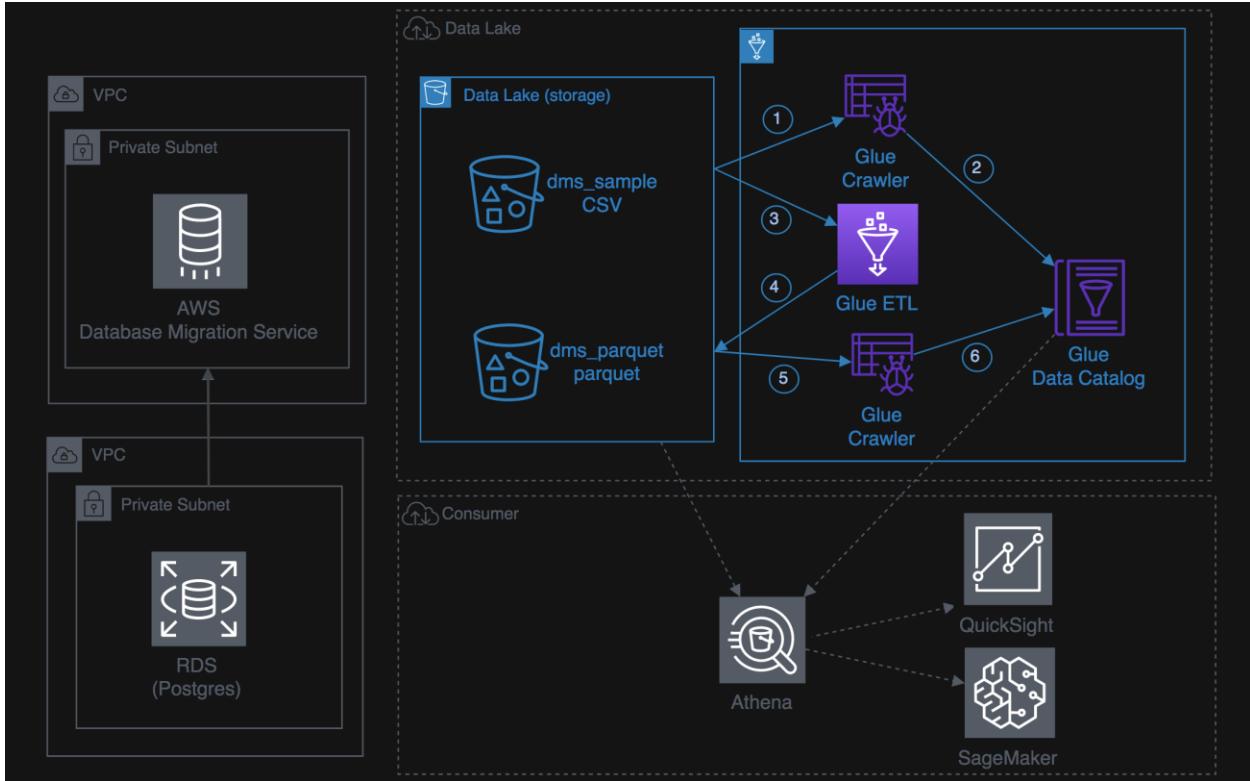
Lab 2. ETL with AWS Glue

Table of Contents

| | |
|---|----|
| <i>Introduction</i> | 2 |
| <i>Get Started Using the Lab Environment</i> | 3 |
| <i>PART A: Data Validation and ETL</i> | 6 |
| Create Glue Crawler for initial full load data | 6 |
| Data Validation Exercise..... | 11 |
| Data ETL Exercise | 12 |
| Create Glue Crawler for Parquet Files | 17 |
| <i>PART B: Glue Job Bookmark (Optional):</i> | 21 |
| Step 1: Create Glue Crawler for ongoing replication (CDC Data)..... | 21 |
| Step 2: Create a Glue Job with Bookmark Enabled | 25 |
| Step 3: Create Glue crawler for Parquet data in S3 | 27 |
| Step 4: Generate CDC data and to observe bookmark functionality | 31 |
| <i>PART C: Glue Workflows (Optional, self-paced)</i> | 32 |
| Overview:..... | 32 |
| Creating and Running Workflows: | 32 |

Introduction

This lab will give you an understanding of the AWS Glue – a fully managed data catalog and ETL service



Prerequisites

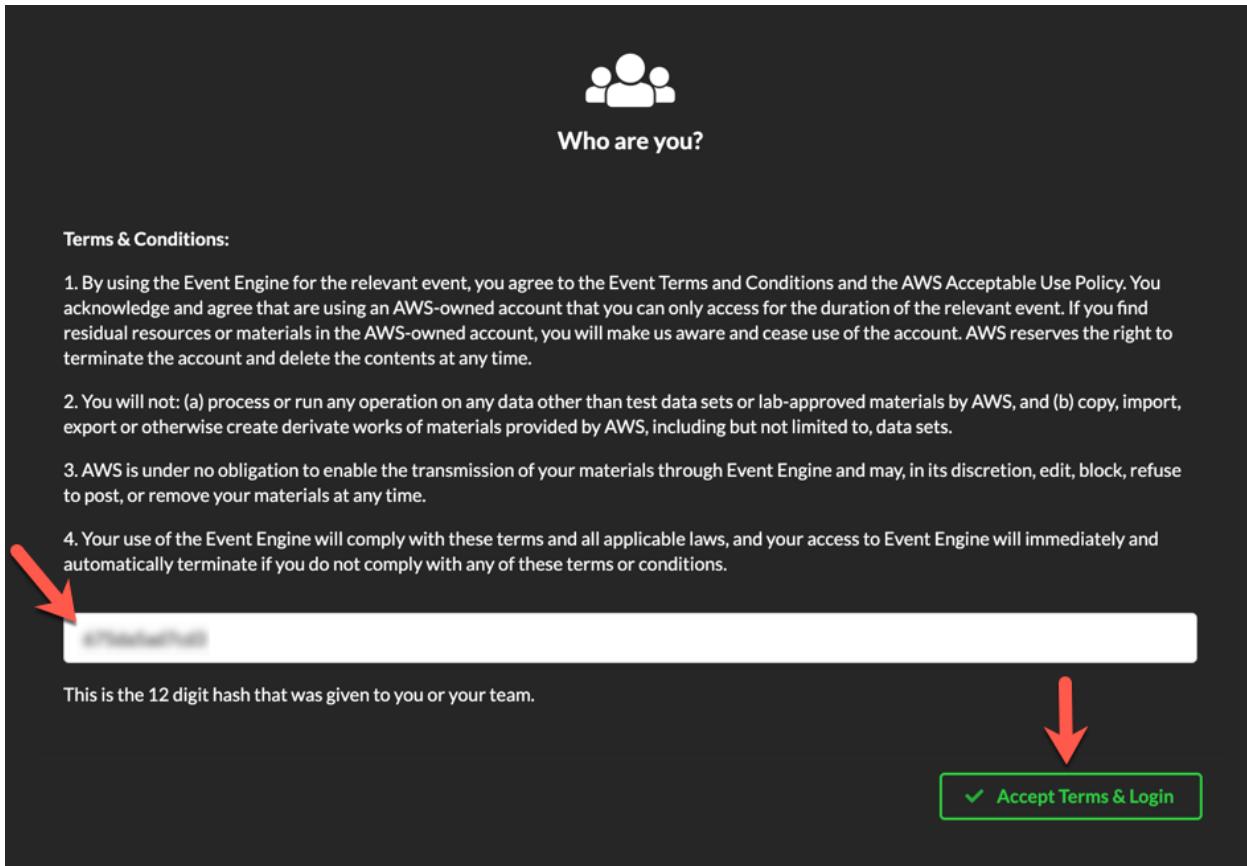
1. Completed Lab 1. Hydrating the Data Lake with DMS

Get Started Using the Lab Environment

Today, you are attending a formal event and you will have been sent your access details beforehand. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions on GitHub - <https://github.com/aws-samples/data-engineering-for-aws-immersion-day>.

A 12-character access code (or 'hash') is the access code that grants you permission to use a dedicated AWS account for the purposes of this workshop.

1. Go to <https://dashboard.eventengine.run/>, enter the access code and click Proceed:



2. On the Team Dashboard web page you will see a set of connection strings and parameters that you will need during the labs. Best to save them to a text file locally, alternatively you can always go to this page to review them. Replace the parameters with the corresponding values from here where indicated in subsequent labs:

Because you're at a formal event, some AWS resources have been pre-deployed for your convenience, for example

- RDS Postgres database that you will use as your source endpoint (parameter **DMSInstanceEndpoint**)

Lab 2. ETL with AWS Glue

Modules

DMS_Student_Prereqs

Outputs:

Data Engineering Workshop

| Parameter | Value |
|----------------------------|--|
| BucketName | mod-08b80667356c4f8a-dmslabs3bucket-lijtekvr232zk |
| BusinessAnalystUser | mod-08b80667356c4f8a-BusinessAnalystUser-1DPVYKJ8GOJK3 |
| BusinessAnalystUserPolicy | BusinessAnalystUserPolicy |
| DMSLabRoleS3 | arn:aws:iam::433083714985:role/mod-08b80667356c4f8a-DMSLabRoleS3-1OV87K4LU3P66 |
| GlueLabRole | mod-08b80667356c4f8a-GlueLabRole-HBJL2G7U4DU8 |
| S3BucketWorkgroupA | mod-08b80667356c4f8a-s3bucketworkgroupa-1sw7181wwqp60 |
| S3BucketWorkgroupB | mod-08b80667356c4f8a-s3bucketworkgroupb-10cz7ir988eho |
| WorkgroupManagerUser | mod-08b80667356c4f8a-WorkgroupManagerUser-1DSHJDROQWRMZ |
| WorkgroupManagerUserPolicy | WorkgroupManagerUserPolicy |

DMS_Instructor_Prereqs

Outputs:

Data Source for DMS Lab

| Parameter | Value |
|---------------------|--|
| DMSInstanceEndpoint | dmslabinstance.ckyqv1sdkm8m.us-east-1.rds.amazonaws.com |
| CDCFunction | arn:aws:lambda:us-east-1:433083714985:function:GenerateCDCData |

3. On the Team Dashboard, please click AWS Console to log into the AWS Management Console:

Team Dashboard

Event

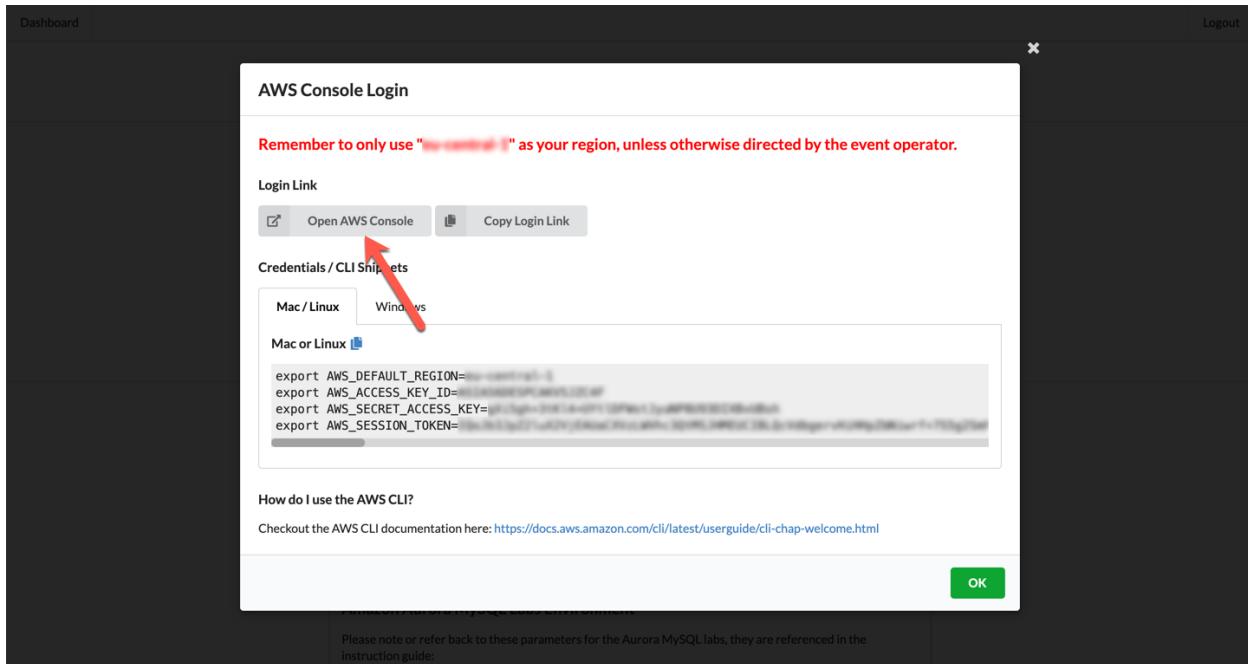
AWS Console SSH Key

Event: Macquarie Bank Data Engineering Immersion Day - Test
Team Name: Igor Izotov

Event ID: d2302d4ae9ff4ea2857846b74f7de7e2
Team ID: 1c2f7ad7ec044b0b8276f917c5983133

4. Click Open Console. For the purposes of this workshop, you will not need to use command line and API access credentials:

Lab 2. ETL with AWS Glue



Once you have completed these steps, you can continue with the rest of this lab

PART A: Data Validation and ETL

Create Glue Crawler for initial full load data

1. Navigate to the AWS Glue service: <https://console.aws.amazon.com/glue/home?region=us-east-1>

The screenshot shows the AWS services console. In the top navigation bar, there is a search bar with the text "glue" typed into it. Below the search bar, a list of services is displayed, with "AWS Glue" being the first item. Other visible items in the list include "AWS Lake Formation" and "S3". At the bottom of the screen, there are links for "All services" and "AWS Glue".

2. On the AWS Glue menu, select **Crawlers**.

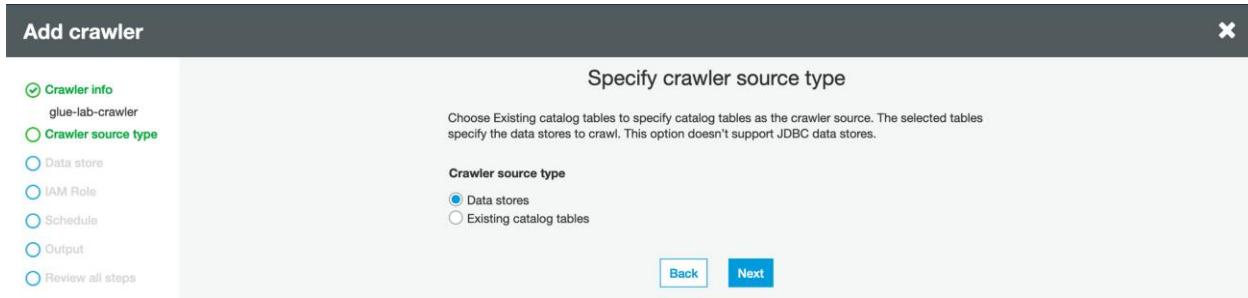
The screenshot shows the "Crawlers" page within the AWS Glue service. The left sidebar has a "Crawlers" section selected. The main area displays a table with columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. A message at the bottom states "You don't have any crawlers yet." with a blue "Add crawler" button below it.

3. Click **Add crawler**.
4. Enter **glue-lab-crawler** as the crawler name for initial data load.
5. Optionally, enter the description. This should also be descriptive and easily recognized and Click **Next**.

The screenshot shows the "Add crawler" wizard. The left sidebar lists steps: "Crawler info" (selected), "Crawler source type", "Data store", "IAM Role", "Schedule", "Output", and "Review all steps". The main area is titled "Add information about your crawler" and contains a "Crawler name" input field with the value "glue-lab-crawler". Below the input field is a note: "Tags, description, security configuration, and classifiers (optional)". A "Next" button is located at the bottom right.

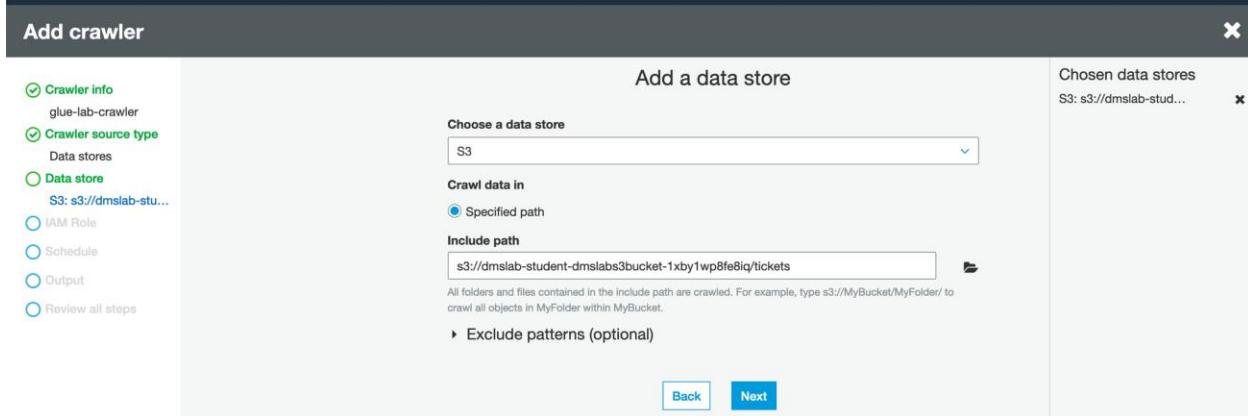
6. Choose **Crawler Source Type** as **Data Stores** and Click **Next**

Lab 2. ETL with AWS Glue

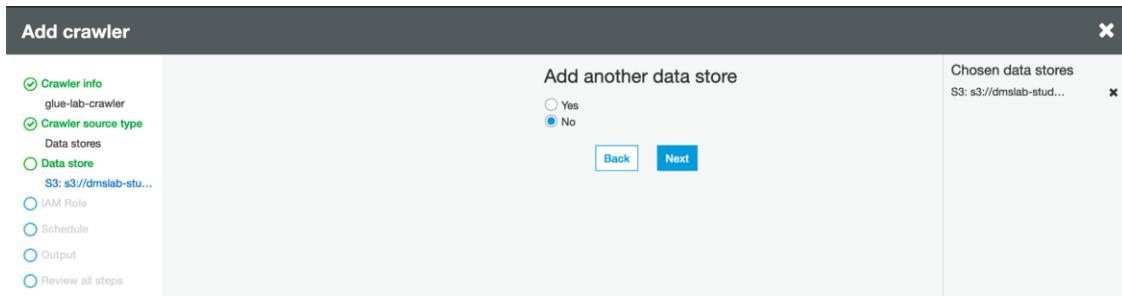


7. On the **Add a data store** page, make the following selections:
 - a. For Choose a data store, click the drop-down box and select **S3**.
 - b. For Crawl data in, select **Specified path in my account**.
 - c. For Include path, browse to the target folder for your DMS initial export from Lab 1, e.g., **s3://dmslab-student-dmslabs3bucket-wot14bf73cw3/tickets**

8. Click **Next**.



9. On the **Add another data store** page, select **No**. and Click **Next**.



10. On the **Choose an IAM role** page, make the following selections:
 - a. Select **Choose an existing IAM role**.
 - b. For **IAM role**, select <stackname>-GlueLabRole-<RandomString> pre-created for you.
For example “dmslab-student-GlueLabRole-ZOQDII7JTBUM”

11. Click **Next**.

Lab 2. ETL with AWS Glue

Add crawler

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)
dmslab-student-GlueLabRole-ZOQDII7JTBUM

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

• s3://dmslab-student-dmslabs3bucket-wot4bf73cw3

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

12. On the Create a schedule for this crawler page, for Frequency, select **Run on demand** and Click **Next**.

Add crawler

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule
Run on demand

Output

Review all steps

Create a schedule for this crawler

Frequency
Run on demand

[Back](#) [Next](#)

13. On the Configure the crawler's output page, click **Add database** to create a new database for our Glue Catalogue.

Add crawler

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule
Run on demand

Output

Review all steps

Configure the crawler's output

Database [?](#)
Choose a database to contain tables

Add database

Prefix added to tables (optional) [?](#)
Type a prefix added to table names

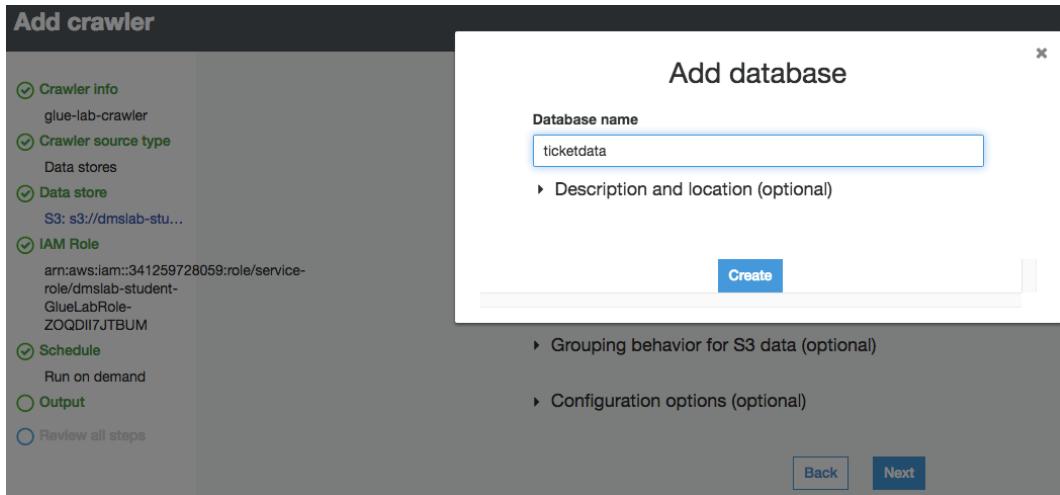
► Grouping behavior for S3 data (optional)

► Configuration options (optional)

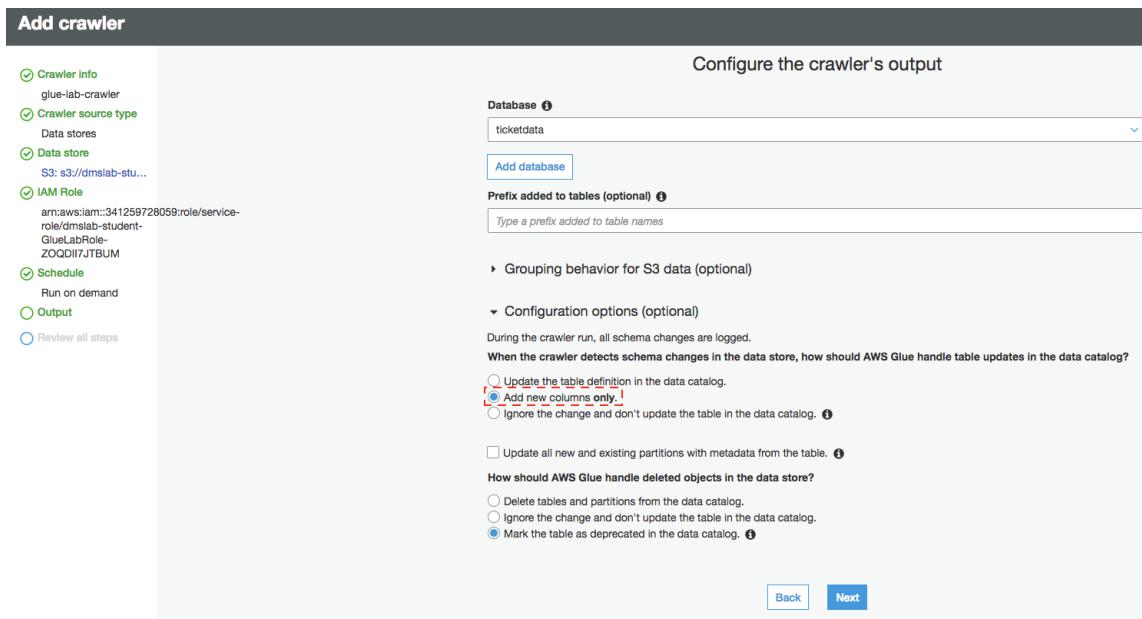
[Back](#) [Next](#)

14. Enter **ticketdata** as your database name and click **create**

Lab 2. ETL with AWS Glue

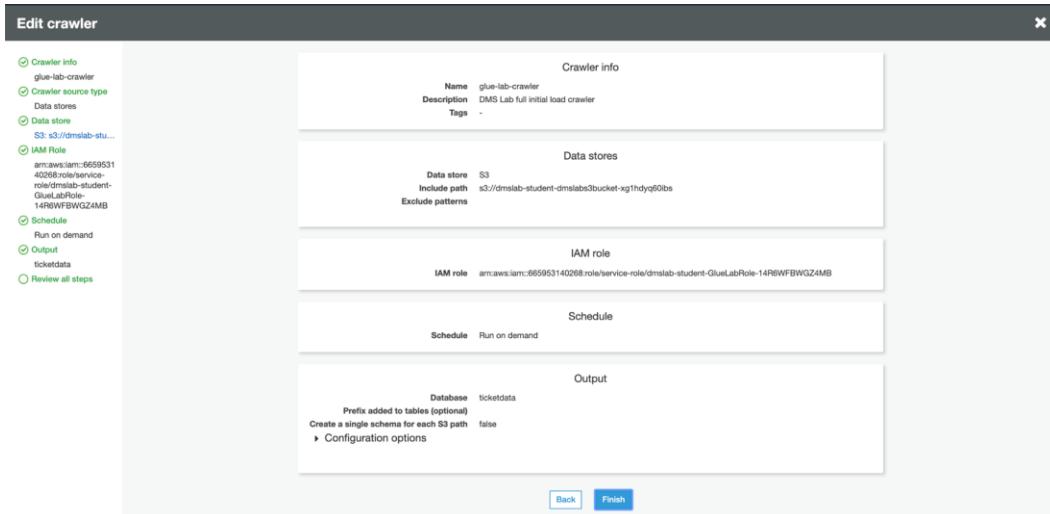


15. For **Prefix added to tables (optional)**, leave the field empty.
16. For **Configuration options (optional)**, select **Add new columns only** and keep the remaining default configuration options and Click **Next**.



17. Review the summary page noting the **Include path** and **Database output** and Click **Finish**. The crawler is now ready to run.

Lab 2. ETL with AWS Glue



18. Click Run it now.

| Name | Schedule | Catalog type | Status | Logs | Last runtime | Median runtime | Tables updated | Tables added |
|------------------|----------|--------------|--------|------|--------------|----------------|----------------|--------------|
| glue-lab-crawler | | Glue | Ready | | 0 secs | 0 secs | 0 | 0 |

Crawler will change status from starting to stopping, wait until crawler comes back to ready state (the process will take a few minutes), you can see that it has created 15 tables.

| Name | Schedule | Catalog type | Status | Logs | Last runtime | Median runtime | Tables updated | Tables added |
|------------------|----------|--------------|--------|------|--------------|----------------|----------------|--------------|
| glue-lab-crawler | | Glue | Ready | Logs | 1 min | 1 min | 0 | 15 |

19. In the AWS Glue navigation pane, click **Databases > Tables**. You can also click the **ticketdata** database to browse the tables.

Lab 2. ETL with AWS Glue

| Name | Database | Location | Classification | Last updated | Deprecated |
|-----------------------|------------|-----------------------------|----------------|-----------------------------|------------|
| mib_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| name_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| nfl_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| nfl_stadium_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| person | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:48 PM ... | |
| player | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| seat | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| seat_type | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| sport_division | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| sport_league | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| sport_location | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| sport_team | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| sporting_event | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| sporting_event_ticket | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |
| ticket_purchase_hist | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM ... | |

Data Validation Exercise

1. Within the Tables section of your **ticketdata** database, click the person table.

| Name | Database | Location | Classification | Last updated | Deprecated |
|------------------|------------|-----------------------------|----------------|-------------------------------|------------|
| mib_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| name_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| nfl_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| nfl_stadium_data | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| person | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:48 PM UTC-5 | |
| player | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| seat | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| seat_type | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| sport_division | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |
| sport_league | ticketdata | s3://dmslab-student-dmsl... | csv | 10 January 2020 1:37 PM UTC-5 | |

You may have noticed that some tables (such as person) have column headers such as col0,col1,col2,col3. In absence of headers or when the crawler cannot determine the header type, default column headers are specified.

This exercise uses the person table in an example of how to resolve this issue.

2. Click **Edit Schema** on the top right side.

Lab 2. ETL with AWS Glue

Tables > person

Last updated 10 Jan 2020 Table Version (Current version) ▾

Table properties

| | | | | | | | | | | | | | |
|-----------------|----------|----------------|------|--------------------|------------------|---------------------------|-----|-------------|---------|-------------------|----|----------------------------------|-----|
| sizeKey | 36656990 | objectCount | 1 | UPDATED_BY_CRAWLER | glue-lab-crawler | CrawlerSchemaSerdeVersion | 1.0 | recordCount | 9164647 | averageRecordSize | 40 | CrawlerSchemaDeserializerVersion | 1.0 |
| compressionType | none | columnsOrdered | true | areColumnsQuoted | false | delimiter | , | typeOfData | file | | | | |

Schema

| Column name | Data type | Partition key | Comment |
|-------------|-----------|---------------|---------|
| 1 col0 | string | | |
| 2 col1 | string | | |
| 3 col2 | string | | |
| 4 col3 | string | | |

3. In the Edit Schema section, double-click **col0** (column name) to open edit mode. Type “id” as the column name.
4. Repeat the preceding step to change the remaining column names to match those shown in the following figure.

Tables > person

Last updated 10 Jan 2020 Table Version (Current version) ▾

Edit schema

| Column name | Data type | Key | Comment |
|--------------|-----------|-----|---------|
| 1 id | string | | |
| 2 full_name | string | | |
| 3 last_name | string | | |
| 4 first_name | string | | |

5. Click **Save**.

Data ETL Exercise

1. In the Glue console, in the left navigation pane, under **ETL** click **Jobs**, and then click **Add job**.

AWS Glue

Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.

User preferences

Action ▾ Filter by attributes

Showing: 0 - 0

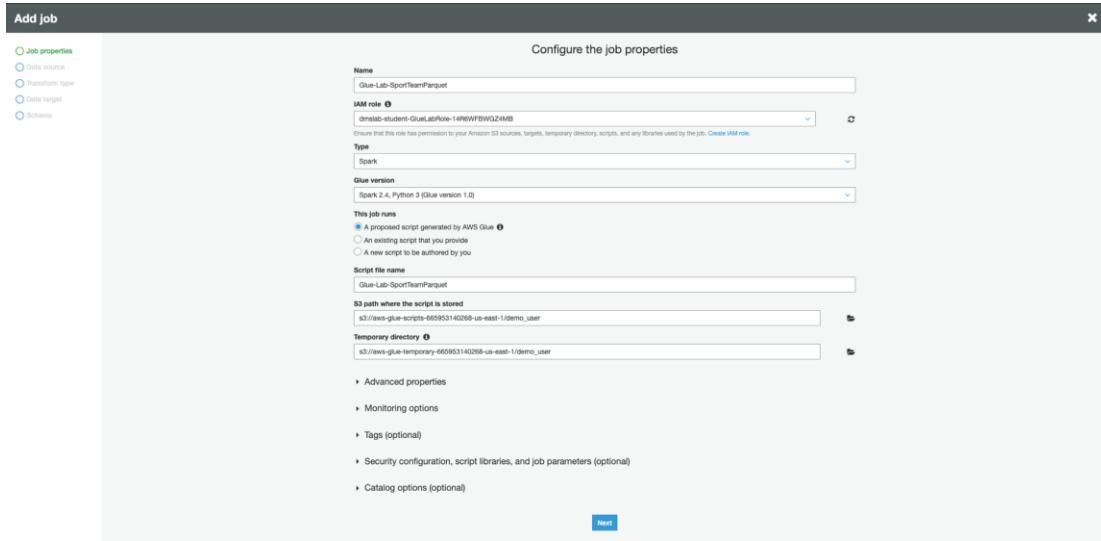
| Name | ETL language | Script location | Last modified | Job bookmark |
|--------------------------------------|--------------|-----------------|---------------|--------------|
| You don't have any jobs defined yet. | | | | |

Add job

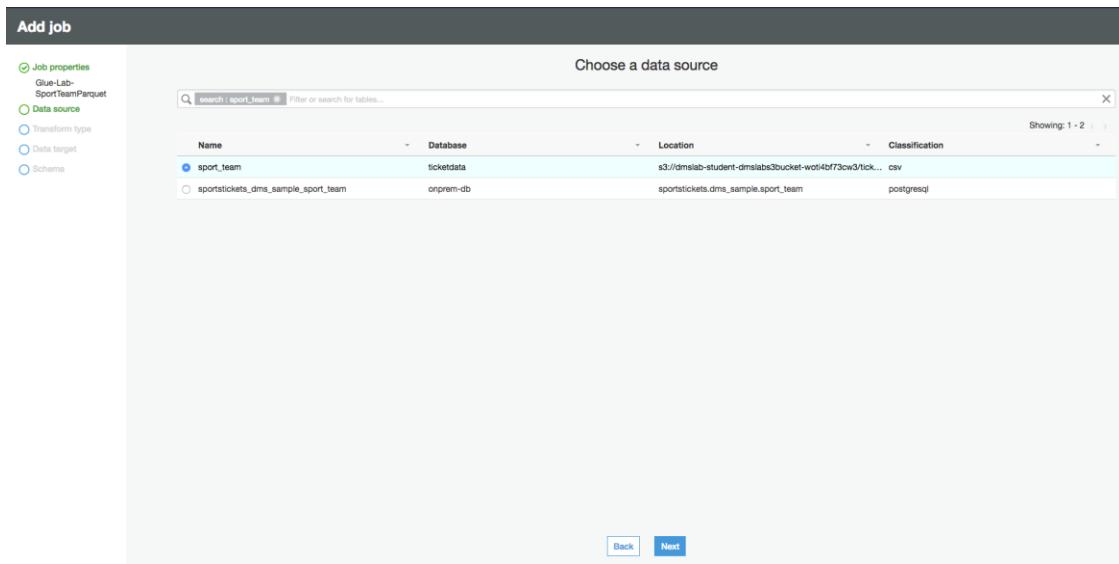
2. On the Job properties page, make the following selections:

Lab 2. ETL with AWS Glue

- a. For **Name**, type “Glue-Lab-SportTeamParquet”
- b. For **IAM role**, choose existing role e.g. “dmslab-student-GlueLabRole-ZOQDII7JTBUM”
- c. For **Type**, Select “Spark”
- d. For **Glue Version**, select “Spark 2.4, Python 3(Glue version 1.0)”
- e. For **This job runs**, select “A proposed script generated by AWS Glue”.
- f. For **Script file name**, type **Glue-Lab-SportTeamParquet**.
- g. For **S3 path where the script is stored**, provide a unique Amazon S3 path to store the scripts. (You can keep the default for this lab.)
- h. For **Temporary directory**, provide a unique Amazon S3 directory for a temporary directory. (You can keep the default for this lab.)

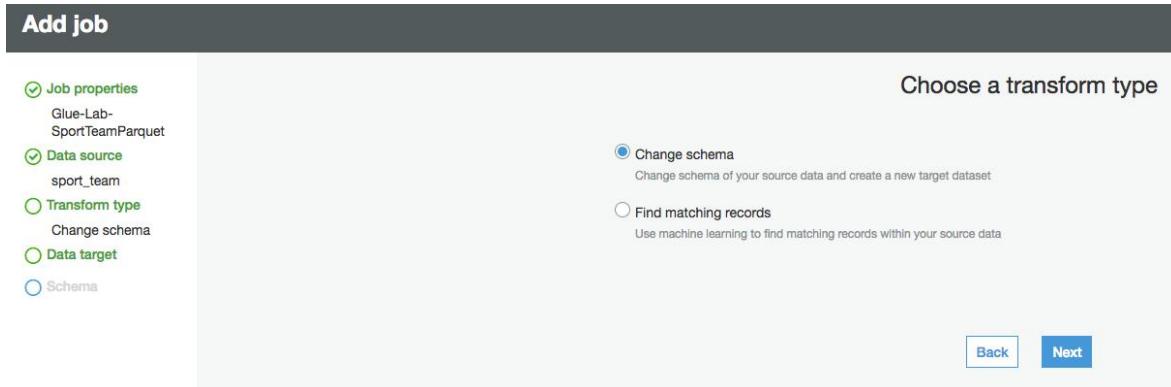


3. Click **Next**
4. On the Choose your data sources page, select **sport_team** and Click **Next**.

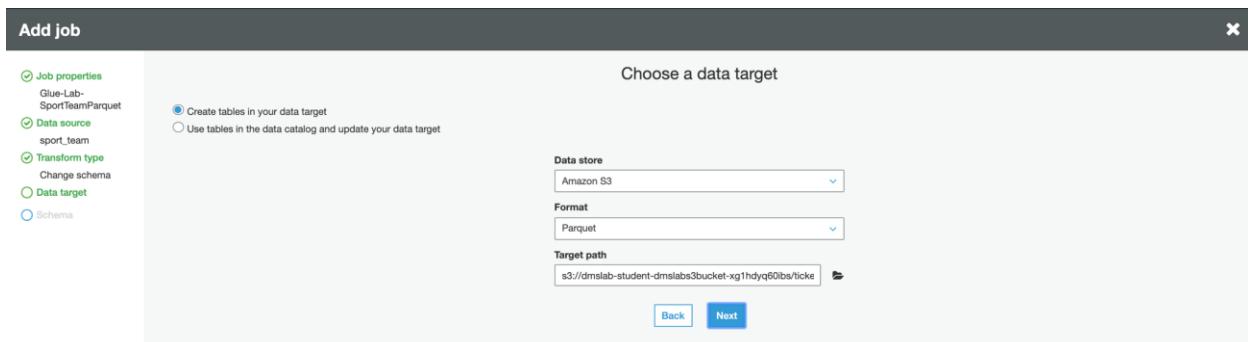


5. On the **Choose a transformation type** page, select **change schema**

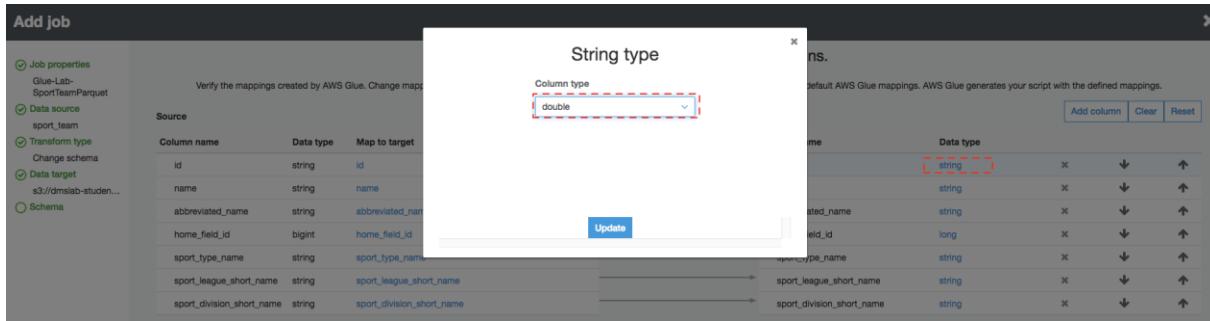
Lab 2. ETL with AWS Glue



6. On the Choose your data targets page, select **Create tables in your data target**.
7. For Data store, select **Amazon S3**.
8. For Format, select **Parquet**.
9. For Target path, choose the s3 bucket and append **/tickets/dms_parquet/sport_team** to it, making the target path look like **s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet/sport_team** – Glue will create necessary folders
10. Click **Next**.



11. Click the target **Data type** to edit the schema mapping for the **id** column. In **String type** pop-up window Select **double** from **Column type** drop down and click **update**.



Lab 2. ETL with AWS Glue

Map the source columns to target columns.

Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

| Source | | Target | | | |
|--|-----------|--|--|-----------|-------|
| Column name | Data type | Map to target | Column name | Data type | |
| <code>id</code> | string | <code>id</code> | <code>id</code> | double | X ↓ ↑ |
| <code>name</code> | string | <code>name</code> | <code>name</code> | string | X ↓ ↑ |
| <code>abbreviated_name</code> | string | <code>abbreviated_name</code> | <code>abbreviated_name</code> | string | X ↓ ↑ |
| <code>home_field_id</code> | bigint | <code>home_field_id</code> | <code>home_field_id</code> | long | X ↓ ↑ |
| <code>sport_type_name</code> | string | <code>sport_type_name</code> | <code>sport_type_name</code> | string | X ↓ ↑ |
| <code>sport_league_short_name</code> | string | <code>sport_league_short_name</code> | <code>sport_league_short_name</code> | string | X ↓ ↑ |
| <code>sport_division_short_name</code> | string | <code>sport_division_short_name</code> | <code>sport_division_short_name</code> | string | X ↓ ↑ |

Add column Clear Reset

Back Save job and edit script

12. Click **Save job and edit script**.

13. View the job. (This screen provides you with the ability to customize this script as required.)

Click **Save** and then **Run Job**.

Job: Glue-Lab-SportTeamParquet Action ▾ Save Run job Generate diagram ⓘ

Insert template at cursor ⓘ Source Target Target Location Transform Spigot ⌂ X

```

1 import sys
2 from awsglue.transforms import *
3 from awsglue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from awsglue.context import GlueContext
6 from awsglue.job import Job
7
8 ## @params: [JOB_NAME]
9 args = getResolvedOptions(sys.argv, ['JOB_NAME'])
10
11 sc = SparkContext()
12 glueContext = GlueContext(sc)
13 spark = glueContext.sparkSession
14 job = Job(glueContext)
15 job.setAppName(args['JOB_NAME'])
16 ## @type: datasource
17 ## @args: [database = "ticketdata", table_name = "sport_team", transformation_ctx = "datasource0"]
18 ## @return: datasource0
19 ## @type: datasource
20 datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "ticketdata", table_name = "sport_team", transformation_ctx = "datasource0")
21 ## @type: ApplyMapping
22 ## @args: [mapping = [{"id": "string", "id": "double"}, {"name": "string", "name": "string"}, {"abbreviated_name": "string", "abbreviated_name": "string"}, {"home_field_id": "long", "home_field_id": "long"}], transformation_ctx = "applymapping1"]
23 ## @type: ApplyMapping
24 ## @args: [frame = applymapping1]
25 applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [{"id": "string", "id": "double"}, {"name": "string", "name": "string"}, {"abbreviated_name": "string", "abbreviated_name": "string"}, {"home_field_id": "long", "home_field_id": "long"}], transformation_ctx = "applymapping1")
26 ## @type: ResolveChoice
27 ## @args: [choice = "make_struct", transformation_ctx = "resolvechoice1"]
28 ## @return: resolvechoice1
29 ## @type: ResolveChoice
30 ## @args: [frame = applymapping1]
31 resolvechoice1 = ResolveChoice.apply(frame = applymapping1, choice = "make_struct", transformation_ctx = "resolvechoice1")
32 ## @type: Transformation
33 dropnullfields3 = DropNullFields.apply(frame = resolvechoice1, transformation_ctx = "dropnullfields3")
34 ## @return: dropnullfields3

```

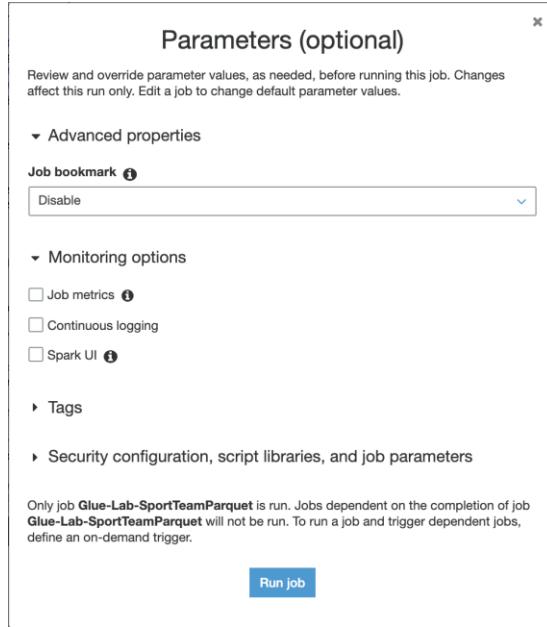
Logs Schema

s3://dmlab-student-dmlabs3bucket/etl-wt4bf73cw3/tickets/dms_parquet/sport_team

14. In **Parameters** option,

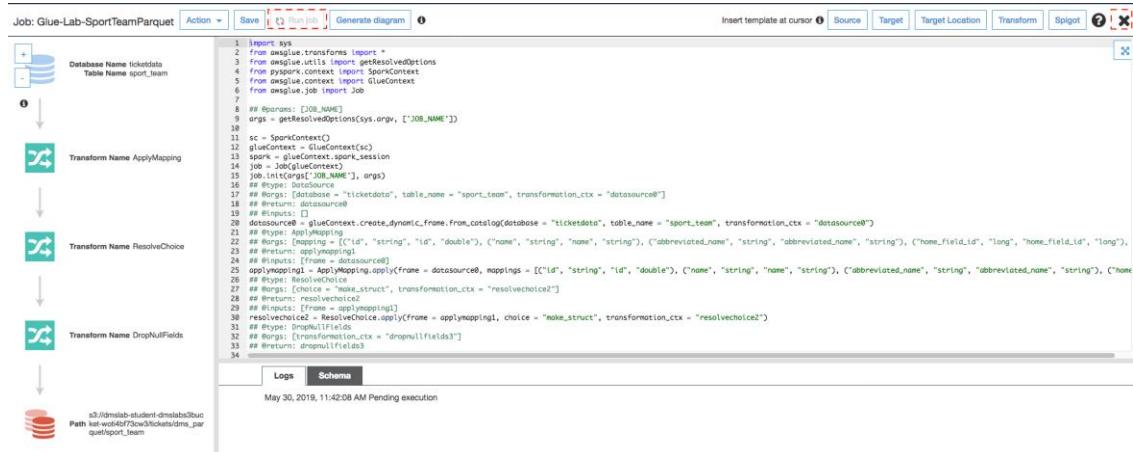
- you can leave **Job bookmark** as **Disable**. AWS Glue tracks data that has already been processed during a previous run of an ETL job by persisting state information from the job run.
- You can leave the **Job metrics** option **Unchecked**. You can collect metrics about AWS Glue jobs and visualize them on the AWS Glue with job metrics.

Lab 2. ETL with AWS Glue



15. Click Run Job

16. You will see job in now running as **Run job** button became greyed out. Click the cross button located in top right corner to close the window to return to the ETL jobs.



17. Click your job to view history and verify that it ran successfully.

| User preferences | | | | | | | | | | |
|---|---------------|-------------------------------|-------------------------|-----------------------------|--------------|---|-----------|--------|-----------|----------------------|
| Add job | Action | Filter by tags and attributes | Showing: 1 - 11 < > ⌂ ⌂ | | | | | | | |
| Name | Type | ETL language | Script location | Last modified | Job bookmark | | | | | |
| Glue-Lab-SportTeamParquet | Spark | python | s3://aws-glue-scri... | 11 March 2020 3:17 PM UTC-7 | Disable | | | | | |
| History Details Script Metrics | | | | | | | | | | |
| View run metrics Rewind job bookmark Showing: 1 - 1 < > ⌂ ⌂ | | | | | | | | | | |
| Run ID | Retry attempt | Run status | Error | Logs | Error logs | Glue version Maximum capacity Triggered by Start time End time Execution time Timeout Delay Job run input | | | | |
| jr_e37b56aa03cd3... | - | Running | Logs | Error logs | 1.0 | 5 | 11 Mar... | 0 secs | 2880 mins | s3://aws-glue-tem... |

Lab 2. ETL with AWS Glue

(Optional) If you plan to continue with other labs outside of this event (for example, the [Athena lab](#)), you'll have to complete the rest of this section to create more ETL Jobs to transform additional tables to parquet, changing their schema as per instructions below., otherwise you can proceed to section **Create Glue Crawler for Parquet Files**

To enable us to join data, we will also update the target data types in the schema. If below **Table1** is indicating need Schema changes as “Yes”. Refer **Table 2** find out column which need to changes with source and target data type during ETL job creation.

Table 1:

| Job Name & Script Filename | Source Table | S3 Target Path | Need Schema Change? |
|-------------------------------------|-----------------------|---|---------------------|
| Glue-Lab-SportLocationParquet | sport_location | s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet/sport_location | No |
| Glue-Lab-SportingEventParquet | sporting_event | s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet/sporting_event | Yes |
| Glue-Lab-SportingEventTicketParquet | sporting_event_ticket | s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet/sporting_event_ticket | Yes |
| Glue-Lab-PersonParquet | person | s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet/person | Yes |

Table 2:

| Job Name | Table | Column | Source Data Type | Target Data Type |
|-------------------------------------|-----------------------|-------------------|------------------|------------------|
| Glue-Lab-SportingEventParquet | sporting_event | start_date_time | STRING | TIMESTAMP |
| Glue-Lab-SportingEventParquet | sporting_event | start_date | STRING | DATE |
| Glue-Lab-SportingEventTicketParquet | sporting_event_ticket | id | STRING | DOUBLE |
| Glue-Lab-SportingEventTicketParquet | sporting_event_ticket | sporting_event_id | STRING | DOUBLE |
| Glue-Lab-SportingEventTicketParquet | sporting_event_ticket | ticketholder_id | STRING | DOUBLE |
| Glue-Lab-PersonParquet | person | id | STRING | DOUBLE |

Once these jobs have completed, we can create a crawler to index these parquet files.

Create Glue Crawler for Parquet Files

1. In the AWS Glue navigation menu, click **Crawlers**, and then click **Add crawler**.

The screenshot shows the AWS Glue interface with the 'Crawlers' tab selected. At the top, there's a search bar with 'Name: glue-lab-crawler' and a filter 'Filter or search for crawlers...'. Below the search bar, there are buttons for 'Run crawler' and 'Action'. A red box highlights the 'Add crawler' button. The main area displays a table with columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. One row is listed: 'glue-lab-crawler' with 'Ready' status, 'Logs' link, '1 min' runtime, '0' updated tables, and '15' added tables. On the left sidebar, there are links for 'Data catalog', 'Tables', 'Connections', 'Classifiers', and 'Settings'.

2. For **Crawler name**, type “glue-lab-parquet-crawler” and Click **Next**.

Lab 2. ETL with AWS Glue

Add crawler

Add information about your crawler

Crawler name: glue-lab-parquet-crawler

Tags, description, security configuration, and classifiers (optional)

Next

Left sidebar:

- Crawler info: glue-lab-parquet-crawler
- Crawler source type:** Data stores
- Data store: S3: s3://dmslab-stu...
- IAM Role
- Schedule
- ...

3. In next screen **Specify crawler source type**, select **Data Source** as choice for **Crawler resource type** and click **Next**.
4. In Add a data store screen
 - a. For **Choose a data store**, select “S3”.
 - b. For **Crawl data in**, select “Specified path in account”.
 - c. For **Include path**, specify the S3 Path (Parent Parquet folder) that contains the nested parquet files e.g., s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet
 - d. Click **Next**.

Add a data store

Choose a data store: S3

Crawl data in:
 Specified path in my account
 Specified path in another account

Include path: s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_parquet

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Exclude patterns (optional)

Back Next

5. For Add another data store, select **No** and Click **Next**.

Add crawler

Add another data store:
 Yes
 No

Chosen data stores: S3: s3://dmslab-stud...

Left sidebar:

- Crawler info: glue-lab-parquet-crawler
- Crawler source type:** Data stores
- Data store: S3: s3://dmslab-stu...
- IAM Role

6. On the Choose an IAM role page, select **Choose an existing IAM role**.
For IAM role, select the existing role “xxx-GlueLabRole-xxx” and Click **Next**.

Lab 2. ETL with AWS Glue

Add crawler

Crawler info
glue-lab-parquet-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role

Choose an existing IAM role

Create an IAM role

IAM role [?](#)
dmslab-student-GlueLabRole-14R6WFBWGZ4MB

This role must provide permissions similar to the AWS managed policy, [AWSGlueServiceRole](#), plus access to your data stores.

- s3://dmslab-student-dmslabs3bucket-xg1hdq60bs/tickets/dms_parquet

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

7. For **Frequency**, select “Run On Demand” and Click **Next**.

Add crawler

Crawler info
glue-lab-parquet-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule

Output

Review all steps

Create a schedule for this crawler

Frequency

Run on demand

[Back](#) [Next](#)

8. For the crawler’s output database, choose your existing database which you created earlier e.g. “ticketdata”
9. For the **Prefix added to tables** (optional), type “parquet_”

Add crawler

Crawler info
glue-lab-parquet-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule
Run on demand

Output

Review all steps

Configure the crawler’s output

Database [?](#)
ticketdata

[Add database](#)

Prefix added to tables (optional) [?](#)
parquet_

▶ Grouping behavior for S3 data (optional)

▶ Configuration options (optional)

[Back](#) [Next](#)

10. Review the summary page and click **Finish**.

Lab 2. ETL with AWS Glue

Add crawler

Crawler info
glue-lab-parquet-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::865953140268:role/service-role/dmslab-student-GlueLabRole-14R6WFBWGZ4MB

Schedule
Run on demand

Output
ticketdata

Review all steps

Crawler info

Name: glue-lab-parquet-crawler
Tags: -

IAM role

IAM role: arn:aws:iam::865953140268:role/service-role/dmslab-student-GlueLabRole-14R6WFBWGZ4MB

Schedule

Schedule: Run on demand

Output

Database: ticketdata
Prefix added to tables (optional): parquet_
Create a single schema for each S3 path: false
▶ Configuration options

[Back](#) [Finish](#)

11. On the notification bar, click **Run it now**. Once your crawler has finished running, you should report that tables were added, 1 to 5, depending on how many parquet ETL conversions you set up in the previous section

AWS Glue

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler "glue-lab-parquet-crawler" completed and made the following changes: 5 tables created, 0 tables updated. See the tables created in database ticketdata.

[User preferences](#)

| Name | Schedule | Catalog type | Status | Logs | Last runtime | Median runtime | Tables updated | Tables added |
|----------------------------|----------|--------------|--------|------|--------------|----------------|----------------|--------------|
| glue-lab-cdc-cra... | | Glue | Ready | Logs | 1 min | 1 min | 0 | 2 |
| glue-lab-crawler | | Glue | Ready | Logs | 1 min | 1 min | 0 | 15 |
| glue-lab-parquet... | | Glue | Ready | Logs | 1 min | 1 min | 0 | 5 |

Confirm you can see the tables:

1. In the left navigation pane, click **Tables**.
2. Add the filter "parquet" to return the newly created tables.

AWS Services Resource Groups IAM Athena Lambda S3 AWS Glue CloudFormation

AWS Glue

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

| Name | Database | Location | Classification | Last updated | Deprecated |
|--------------------------------------|------------|---|----------------|-------------------------------|------------|
| mb_data | ticketdata | s3://dmslab-student-dmslabs3buck... csv | | 10 January 2020 1:37 PM UTC-5 | |
| name_data | ticketdata | s3://dmslab-student-dmslabs3buck... csv | | 10 January 2020 1:37 PM UTC-5 | |
| nfl_data | ticketdata | s3://dmslab-student-dmslabs3buck... csv | | 10 January 2020 1:37 PM UTC-5 | |
| nfl_stadium_data | ticketdata | s3://dmslab-student-dmslabs3buck... csv | | 10 January 2020 1:37 PM UTC-5 | |
| parquet_person | ticketdata | s3://dmslab-student-dmslabs3buck... parquet | | 23 January 2020 1:49 PM UTC-5 | |
| parquet_person_ticket | ticketdata | s3://dmslab-student-dmslabs3buck... parquet | | 23 January 2020 1:49 PM UTC-5 | |
| parquet_sport_team | ticketdata | s3://dmslab-student-dmslabs3buck... parquet | | 23 January 2020 1:49 PM UTC-5 | |
| parquet_sporting_event | ticketdata | s3://dmslab-student-dmslabs3buck... parquet | | 23 January 2020 1:49 PM UTC-5 | |
| parquet_sporting_event_ticket | ticketdata | s3://dmslab-student-dmslabs3buck... parquet | | 23 January 2020 1:49 PM UTC-5 | |
| person | ticketdata | s3://dmslab-student-dmslabs3buck... csv | | 10 January 2020 1:48 PM UTC-5 | |

PART B: Glue Job Bookmark (Optional):

****Pre-requisite: Completion of CDC part of Lab 1****

Step 1: Create Glue Crawler for ongoing replication (CDC Data)

Now, let's repeat this process to load the data from change data capture.

1. On the AWS Glue menu, select Crawlers.

2. Click **Add crawler**.
3. Enter the crawler name for ongoing replication. This name should be descriptive and easily recognized (e.g., "glue-lab-cdc-crawler").
4. Optionally, enter the description. This should also be descriptive and easily recognized and Click **Next**.

5. Choose **Crawler Source Type** as **Data Stores** and Click **Next**

6. On the Add a data store page, make the following selections:
 - a. For **Choose a data store**, click the drop-down box and select **S3**.
 - b. For **Crawl data in**, select **Specified path in my account**.
 - c. For **Include path**, enter the **target folder** for your DMS ongoing replication, e.g., "**s3://xxx-dmslabs3bucket-xxx/cdc/dms_sample**"
7. Click **Next**.

Lab 2. ETL with AWS Glue

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store

IAM Role

Schedule

Output

Review all steps

Add a data store

Choose a data store
S3

Crawl data in

Specified path in my account
 Specified path in another account

Include path
s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/cdc/dms_sample

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Exclude patterns (optional)

[Back](#) [Next](#)

8. On the **Add another data store page**, select **No** and Click **Next**.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role

Schedule

Output

Review all steps

Add another data store

Chosen data stores
S3: s3://dmslab-stu...

Yes
 No

[Back](#) [Next](#)

9. On the **Choose an IAM role** page, make the following selections:

- Select **Choose an existing IAM role**.
- For **IAM role**, select **xxx-GlueLabRole-xxx**. E.g. “dmslab-student-GlueLabRole-ZOQDII7JTBUM”

10. Click **Next**.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role

dmslab-student-GlueLabRole-14R6WFBWGZ4MB

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

- s3://dmslab-student-dmslabs3bucket-xg1hydq60lbs/cdc/dms_sample

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

11. On the **Create a schedule for this crawler** page, for Frequency, select **Run on demand** and Click **Next**.

Lab 2. ETL with AWS Glue

Add crawler

Create a schedule for this crawler

Frequency: Run on demand

Back Next

Crawler info: glue-lab-cdc-crawler

Crawler source type: Data stores

S3: s3://dmslab-stu...

IAM Role: arn:aws:iam::6659531:40268:role/service-role/dmslab-student-GlueLabRole-14R6WFWBWZ4MB

Schedule: Run on demand

Output: ticketdata

Review all steps

12. On the Configure the crawler's output page, select the existing **Database** for crawler output (e.g., "ticketdata").
13. For **Prefix added to tables (optional)**, specify "cdc_"
14. For Configuration options (optional), keep the default selections and click **Next**.

Add crawler

Configure the crawler's output

Database: ticketdata

Add database

Prefix added to tables (optional): cdc_

Type a prefix added to table names

Grouping behavior for S3 data (optional)

Configuration options (optional)

When the crawler detects schema changes in the data store, how should AWS Glue handle table updates in the data catalog?

- Update the table definition in the data catalog.
- Add new columns only.
- Ignore the change and don't update the table in the data catalog. ⓘ

Update all new and existing partitions with metadata from the table. ⓘ

How should AWS Glue handle deleted objects in the data store?

- Delete tables and partitions from the data catalog.
- Ignore the change and don't update the table in the data catalog.
- Mark the table as deprecated in the data catalog. ⓘ

Back Next

Crawler info: glue-lab-cdc-crawler

Crawler source type: Data stores

S3: s3://dmslab-stu...

IAM Role: arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZQDQII7JTBUM

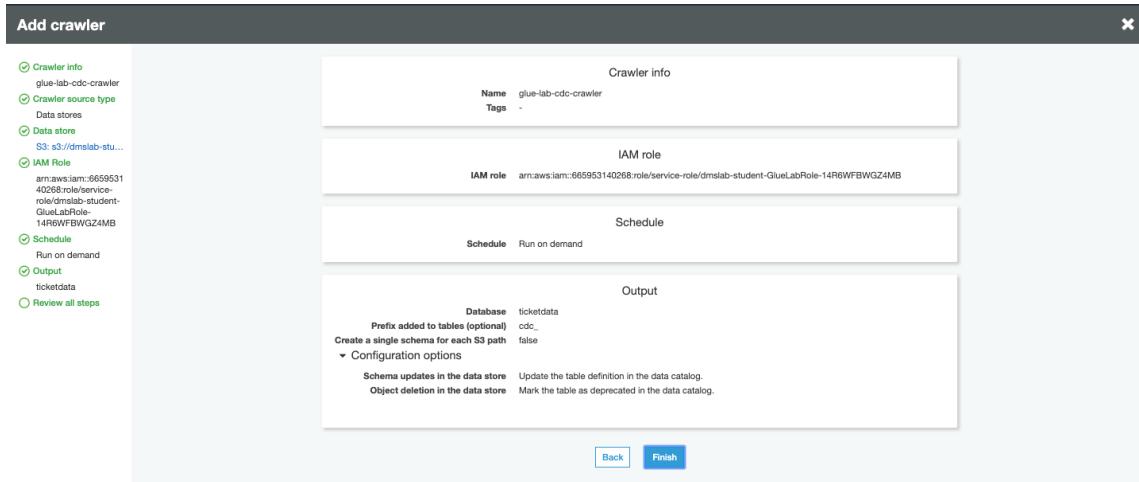
Schedule: Run on demand

Output: ticketdata

Review all steps

15. Review the summary page noting the Include path and Database target and Click **Finish**. The crawler is now ready to run.

Lab 2. ETL with AWS Glue



16. Click Run it now.

AWS Glue

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler glue-lab-cdc-crawler was created to run on demand. [Run it now!](#)

| Name | Schedule | Catalog type | Status | Logs | Last runtime | Median runtime | Tables updated | Tables added |
|---------------------|----------|--------------|--------|--------|--------------|----------------|----------------|--------------|
| glue-lab-cdc-cra... | Glue | Ready | 0 secs | 0 secs | 0 | 0 | 0 | 0 |
| glue-lab-crawler | Glue | Ready | Logs | 1 min | 1 min | 0 | 0 | 15 |

17. When the crawler is completed, you can see it has “Status” as **Ready**, Crawler will change status from starting to stopping, wait until crawler comes back to ready state, you can see that it has created 2 tables.

AWS Glue

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler "glue-lab-cdc-crawler" completed and made the following changes: 2 tables created, 0 tables updated. See the tables created in database ticketdata.

| Name | Schedule | Catalog type | Status | Logs | Last runtime | Median runtime | Tables updated | Tables added |
|---------------------|----------|--------------|--------|-------|--------------|----------------|----------------|--------------|
| glue-lab-cdc-cra... | Glue | Ready | Logs | 1 min | 1 min | 0 | 0 | 2 |
| glue-lab-crawler | Glue | Ready | Logs | 1 min | 1 min | 0 | 0 | 15 |

18. Click the database name (e.g., "ticketdata") to browse the tables. Specify "cdc" as the filter to list only newly imported tables.

Lab 2. ETL with AWS Glue

AWS Glue

Data catalog

Databases

- Tables
- Connections
- Crawlers
- Classifiers
- Settings

ETL

Workflows

Jobs

ML Transforms

Triggers

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

| Name | Database | Location | Classification | Last updated | Deprecated |
|---------------------------|------------|-------------------------------------|----------------|-------------------------------|------------|
| cdc_sporting_event_ticket | ticketdata | s3://dmslab-student-dmslabs3buck... | csv | 23 January 2020 4:38 PM UTC-5 | |
| cdc_ticket_purchase_hist | ticketdata | s3://dmslab-student-dmslabs3buck... | csv | 23 January 2020 4:38 PM UTC-5 | |
| mlb_data | ticketdata | s3://dmslab-student-dmslabs3buck... | csv | 10 January 2020 1:37 PM UTC-5 | |
| name_data | ticketdata | s3://dmslab-student-dmslabs3buck... | csv | 10 January 2020 1:37 PM UTC-5 | |
| nfl_data | ticketdata | s3://dmslab-student-dmslabs3buck... | csv | 10 January 2020 1:37 PM UTC-5 | |
| nfl_stadium_data | ticketdata | s3://dmslab-student-dmslabs3buck... | csv | 10 January 2020 1:37 PM UTC-5 | |
| parquet_person | ticketdata | s3://dmslab-student-dmslabs3buck... | parquet | 23 January 2020 1:49 PM UTC-5 | |
| parquet_sport_location | ticketdata | s3://dmslab-student-dmslabs3buck... | parquet | 23 January 2020 1:49 PM UTC-5 | |
| parquet_sport_team | ticketdata | s3://dmslab-student-dmslabs3buck... | parquet | 23 January 2020 1:49 PM UTC-5 | |

Save view Showing: 1 - 22

Step 2: Create a Glue Job with Bookmark Enabled

- On the left-hand side of Glue Console, click on Jobs and then Click on Add Job

AWS Glue

Data catalog

Databases

Tables

Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.

| Name | Type | ETL language | Script location | Last modified | Job bookmark |
|------|------|--------------|-----------------|---------------|--------------|
|------|------|--------------|-----------------|---------------|--------------|

Add job Action Filter by tags and attributes User preferences Showing: 1 - 9

- On the Job properties page, make the following selections:
 - For **Name**, type **Glue-Lab-TicketHistory-Parquet-with-bookmark**.
 - For **IAM role**, choose existing role “xxx-GlueLabRole-xxx”
 - For **Type**, Select **Spark**
 - For **Glue Version**, select **Spark 2.4, Python 3 (Glue version 1.0)**
 - For **This job runs**, select **A proposed script generated by AWS Glue**.
 - For **Script file name**, type **Glue-Lab-TicketHistory-Parquet-with-bookmark**.
 - For **S3 path where the script is stored**, provide a unique Amazon S3 path to store the scripts. (You can keep the default for this lab.)
 - For **Temporary directory**, provide a unique Amazon S3 directory for a temporary directory. (You can keep the default for this lab.)
- Expand the **Advanced properties** section. For Job bookmark, select **Enable** from the drop-down option.
- Expand on the **Monitoring** options, enable **Job metrics**.
- Click **Next**

Add job

Configure the job properties

Job properties

Job properties

- Glue-Lab-TicketHistory-Parquet-with-bookmark
- Data source
- Transform type
- Data target
- Schema

Name

IAM role
 GlueLabRole-14N9WFBWGZ4MB

Ensures that this role has permission to access your Amazon S3 sources, targets, temporary directories, scripts, and any libraries used by the job. Create IAM role.

Type

Glue version
 Spark 2.4, Python 3 (Glue Version 1.0)

Spark 2.4, Python 3 (Glue Version 1.0)

This job runs

A proposed script generated by AWS Glue
 An existing script that you provide
 A new script to be authored by you

Script file name

S3 path where the script is stored

Temporary directory
 s3://aws-glue-temporary-665953140268-us-east-1/destdir1

s3://aws-glue-temporary-665953140268-us-east-1/destdir1

Advanced properties

Job bookmark
 Enable

Job bookmark

Monitoring options

Job metrics
 Continuous logging
 Spark UI

Tags (optional)

Lab 2. ETL with AWS Glue

6. In Choose a data source, select **cdc_ticket_purchase_hist** as we are generating new data entries for **ticket_purchase_hist** table. Click **Next**

| Name | Database | Location | Classification |
|--|----------------|--|----------------|
| bookmark_parquet_ticket_purchase_history | ticketdata | s3://dmslab-student-dmlabs3bucket-xg1hydg0bs/cdc_bookmark/ticket... | parquet |
| cdc_sporting_event_ticket | ticketdata | s3://dmslab-student-dmlabs3bucket-xg1hydg0bs/cdc/dms_sample/sport... | csv |
| cdc_ticket_purchase_hist | ticketdata | s3://dmslab-student-dmlabs3bucket-xg1hydg0bs/cdc/dms_sample/tic... | csv |
| clickstream_data | processed-data | s3://rawdataset-decentralizedClickstream_data/ | json |
| csv_clickstream_data | processed-data | s3://processed-decentralizedClickstream_data/ | csv |

7. In Choose a transform type, select **Change Schema** and Click **Next**

Choose a transform type

- Change schema
- Find matching records

Change schema: Change schema of your source data and create a new target dataset.

Find matching records: Use machine learning to find matching records within your source data.

Back Next

8. In Choose a data target:

- For **Data store**: select **amazon S3**
- Format: **parquet**
- Target path**: `s3://xxx-dmslabs3bucket-xxx/cdc_bookmark/ticket_purchase_history_data/`
- Click **Next**

Choose a data target

Create tables in your data target

Use tables in the data catalog and update your data target

Data store: Amazon S3

Format: Parquet

Target path: 1hydg0bs/cdc_bookmark/ticket_purchase_history/data/

Back Next

9. In map the source columns to target columns window, leave everything default and Click on **Save job and edit script**.

Map the source columns to target columns.

Verify the mappings created by AWS Glue. Change mappings by choosing other columns with Map to target. You can Clear all mappings and Reset to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

| Source | Column name | Data type | Map to target | Target | Column name | Data type |
|--------|--------------------------|-----------|--------------------------|--------|--------------------------|-----------|
| | op | string | op | | op | string |
| | sporting_event_ticket_id | string | sporting_event_ticket_id | | sporting_event_ticket_id | string |
| | purchased_by_id | string | purchased_by_id | | purchased_by_id | string |
| | transaction_date_time | string | transaction_date_time | | transaction_date_time | string |
| | transferred_from_id | string | transferred_from_id | | transferred_from_id | string |
| | purchase_price | double | purchase_price | | purchase_price | double |

Add column Clear Reset

Back Save job and edit script

Lab 2. ETL with AWS Glue

10. In the next window, review the job script and click on **Run job**. Click on close mark on the top right of the window to close the screen.

```

Job: Glue-Lab-TicketHistory-Parquet-with-bookmark
Action ▾ Save Run job Generate diagram ⚙
Database Name ticketdata
Table Name idc_ticket_purchase_hist
Transform Name ApplyMapping
Transform Name ResolveChoice
Transform Name DropNullFields
e3/enable-data-enrichment
Path: s3://dmlab-student-dmslabs3bucket-xg1hdq60ibs
Pattern: s3://dmlab-student-dmslabs3bucket-xg1hdq60ibs/ticket_purchase_hist/

```

```

1 import sys
2 from glue.job import Job
3 from glue.utils import getGlueOptions
4 from pyglue import SparkContext
5 from glue.context import GlueContext
6 from glue.job import Job
7
8 # ---#
9 args = getGlueOptions(sys.argv, ['JOB_NAME'])
10
11 sc = SparkContext()
12 glueContext = GlueContext(sc)
13 spark = glueContext.sparkSession
14 job = Job(glueContext)
15 job.init('glue-' + JOB_NAME, args)
16
17 ## Boring: Database = "ticketdata", table_name = "idc_ticket_purchase_hist", transformation_ctx = "datasource0"
18
19 ## Boring: Datasource0
20 ## Boring: glueContext.create_dynamic_frame.from_catalog(database = "ticketdata", table_name = "idc_ticket_purchase_hist", transformation_ctx = "datasource0")
21 ## Boring: ApplyMapping
22 ## Boring: (["op", "string", "op", "string"], ["reporting_event_ticket_id", "string", "reporting_event_ticket_id", "string"], ["purchased_by_id", "string", "purchased_by_id", "string"], ["transaction_date_time", "string", "transaction_date_time", "string"])
23 ## Boring: [Frame - datasource0]
24 ## Boring: dropnullfields(frame = datasource0, mappings = [{"op": "string", "op": "string", "reporting_event_ticket_id": "string", "reporting_event_ticket_id": "string"}, {"purchased_by_id": "string", "purchased_by_id": "string", "transaction_date_time": "string", "transaction_date_time": "string"}])
25 ## Boring: ResolveChoice
26 ## Boring: (["op", "string", "op", "string"], ["reporting_event_ticket_id", "string", "reporting_event_ticket_id", "string"], ["purchased_by_id", "string", "purchased_by_id", "string"], ["transaction_date_time", "string", "transaction_date_time", "string"])
27 ## Boring: resolvechoice0
28 ## Boring: resolvechoice1
29 ## Boring: resolvechoice2
30 ## Boring: resolvechoice3
31 ## Boring: resolvechoice4
32 ## Boring: resolvechoice5
33 ## Boring: dropnullfields
34 ## Boring: dropnullfields0
35 dropnullfields0 = DropNullFields.apply(frame = resolvechoice2, transformation_ctx = "dropnullfields0")
36 ## Boring: DropNullFields
37 ## Boring: dropnullfields1
38 ## Boring: dropnullfields2
39 ## Boring: dropnullfields3
40 datasource4 = glueContext.write_dynamic_frame.frame_options(frame = dropnullfields3, connection_type = "s3", connection_options = {"path": "s3://dmlab-student-dmslabs3bucket-xg1hdq60ibs/cdc_bookmark/ticket_purchase_history/data"}, format = "parquet", transformation_ctx = "datasink4")
41 job.commit()

```

11. Once the job finishes its run, check the **S3 bucket** for the parquet partitioned data.

Amazon S3 > dmlab-student-dmslabs3bucket-xg1hdq60ibs > cdc_bookmark > ticket_purchase_history > data

Overview

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload + Create folder Download Actions US East (N. Virginia) ▾

| Name | Last modified | Size | Storage class |
|--|----------------------------------|--------|---------------|
| part-00000-49bea7fc-2ac1-4787-b431-9e16f5e24a3f-c0000.snappy.parquet | Jan 24, 2020 7:03:16 PM GMT-0500 | 1.1 MB | Standard |
| part-00001-49bea7fc-2ac1-4787-b431-9e16f5e24a3f-c0000.snappy.parquet | Jan 24, 2020 7:03:16 PM GMT-0500 | 1.2 MB | Standard |

Step 3: Create Glue crawler for Parquet data in S3

- Once you have the data in S3 bucket, navigate to **Glue Console** and now we will crawl the parquet data in S3 to create data catalog.
- Click on **Add crawler**
- In crawler configuration window, provide crawler name as **glue_lab_cdc_bookmark_crawler** and Click **Next**.

Add information about your crawler

Crawler name: glue_lab_cdc_bookmark_crawler

Tags, description, security configuration, and classifiers (optional)

Next

- In specify **crawler source type**, for crawler source type, select **Data stores**. Click **Next**

Lab 2. ETL with AWS Glue

Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type

Data stores

Existing catalog tables

[Back](#) [Next](#)



5. In **Add a data store**:
 - a. For **Choose a data store**, select **S3**
 - b. For the path, provide this: `s3:// xxx-dmslabs3bucket-xxx` and append `/cdc_bookmark/ticket_purchase_history/`.
6. Click on **Next**

Add a data store

Choose a data store

S3

Crawl data in

Specified path

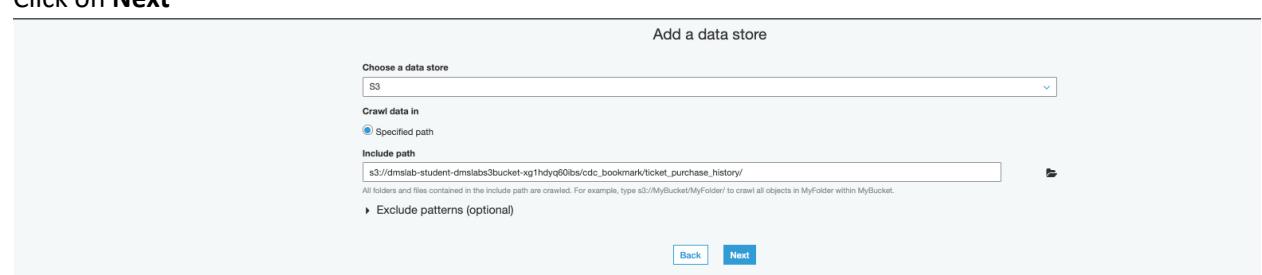
Include path

`s3://dmslab-student-dmslabs3bucket-xg1hdyyq60bs/cdc_bookmark/ticket_purchase_history/`

All folders and files contained in the include path are crawled. For example, type `s3://MyBucket/MyFolder` to crawl all objects in MyFolder within MyBucket.

Exclude patterns (optional)

[Back](#) [Next](#)



7. For **Add another data store**, select **No** and click **Next**.

Add another data store

Yes

No

[Back](#) [Next](#)



8. In **Choose an IAM role**, select **Choose an existing IAM role** and select the role that you created as part of the DMS_Student Lab. (for eg, this role name looks something like this: `dmslab-student-GlueLabRole-<random-alphanumeric-characters>`)

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role

Choose an existing IAM role

Create an IAM role

IAM role 

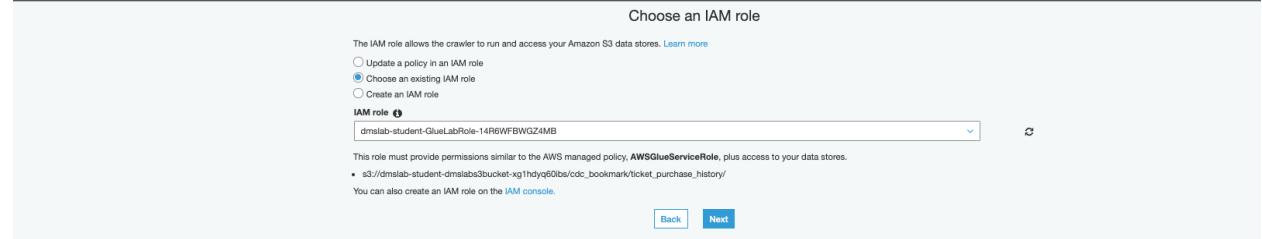
`dmslab-student-GlueLabRole-14RBWFBWQZ4MB`

This role must provide permissions similar to the AWS managed policy, `AWSGlueServiceRole`, plus access to your data stores.

`s3://dmslab-student-dmslabs3bucket-xg1hdyyq60bs/cdc_bookmark/ticket_purchase_history/`

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)



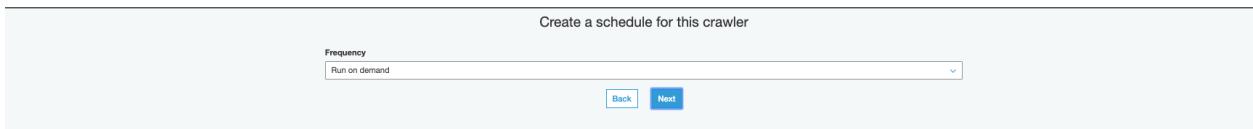
9. For setting the **frequency** in create a schedule for this crawler, select “Run on demand”. Click **Next**

Create a schedule for this crawler

Frequency

`Run on demand`

[Back](#) [Next](#)



10. For the crawler’s output:
 - a. For Database, select “**ticket**” database.
 - b. Optionally, add prefix to the newly created tables for easy identification. Provide the prefix as “**bookmark_parquet_**”
 - c. Click **Next**

Lab 2. ETL with AWS Glue

Configure the crawler's output

Database Add database

Prefix added to tables (optional)

Grouping behavior for S3 data (optional)

Configuration options (optional)

Back Next

11. Review all the details and click on **Finish**. Next, run the crawler.

Crawler info
Name: glue_lab_cdc_bookmark_crawler
Tags: -

Data stores
Data store: S3 s3://dmlab-student-dmlabs3bucket-xg1hdyg60ba/cdc_bookmark/ticket_purchase_history/
Include path: s3://dmlab-student-dmlabs3bucket-xg1hdyg60ba/cdc_bookmark/ticket_purchase_history/
Exclude patterns: -

IAM role
IAM role: arn:aws:iam::665953140268:role/service-role/dmlab-student-GlueLabRole-14RIRWFBWGZAMB

Schedule
Schedule: Run on demand

Output
Database: ticketdata
Prefix added to tables (optional): bookmark_parquet_
Create a single schema for each S3 path: false
Configuration options: -

Back Finish

12. After the crawler finishes running, click on Databases, select “**ticketdata**” and view tables in this database. You will find the newly created table as “**bookmark_parquet_ticket_purchase_history**”

AWS Glue

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

| Name | Database | Location | Classification | Last updated | Deprecated |
|--|------------|---|----------------|-------------------------------|------------|
| bookmark_parquet_ticket_purchase_history | ticketdata | s3://dmlab-student-dmlabs3bucket-xg1hdyg60ba/ | parquet | 24 January 2020 7:14 PM UTC-5 | - |
| cdc_sporting_event_ticket | ticketdata | s3://dmlab-student-dmlabs3bucket-xg1hdyg60ba/ | csv | 24 January 2020 5:13 PM UTC-5 | - |
| cdc_ticket_purchase_hist | ticketdata | s3://dmlab-student-dmlabs3bucket-xg1hdyg60ba/ | csv | 24 January 2020 5:13 PM UTC-5 | - |
| mb_data | ticketdata | s3://dmlab-student-dmlabs3bucket-xg1hdyg60ba/ | csv | 10 January 2020 1:37 PM UTC-5 | - |

13. Once the table is created, click on **Action** and from dropdown select **View Data**.

Because it's the first time we're using Athena in this AWS Account, click **Get Started**



Then click **set up a query result location** in Amazon S3 at the top

A screenshot of the Amazon Athena Settings page. At the top, there are tabs for "Sources" and "Workgroup : primary". A callout box highlights the "Workgroup" tab. Below the tabs, a message says: "Before you run your first query, you need to set up a query result location in Amazon S3. Learn more".

In the pop-up window in the **Query result location** field, enter your s3 bucket location followed by /, so that it looks like **s3://xxx-dmslabs3bucket-xxx/** and click **Save**

A screenshot of the "Settings" pop-up window. It has a close button "X" in the top right corner. The title is "Settings". Below the title, it says "Settings apply by default to all new queries. [Learn more](#)". Under "Workgroup", "primary" is selected. The "Query result location" field is empty and has a placeholder "Example: s3://query-results-bucket/folder/" with an info icon. Below it are two checkboxes: "Encrypt query results" and "Autocomplete", both with info icons. At the bottom are "Cancel" and "Save" buttons.

To select some rows from the table, try running:

```
SELECT * FROM "ticketdata"."bookmark_parquet_ticket_purchase_history_data"  
limit 10;
```

To get a row count, run:

Lab 2. ETL with AWS Glue

```
SELECT count(*) as recordcount FROM  
"ticketdata"."bookmark_parquet_ticket_purchase_history_data";
```

Before moving on to next step, note the rowcount.

Step 4: Generate CDC data and to observe bookmark functionality

Generate more CDC data the same way you did it in Lab 1. You may need to wait 5 to 10 minutes for CDC data to first reflect in your RDS postgres database and then picked up by DMS CDC migration task.

1. To make sure the new data has been successfully generated, check the S3 bucket for cdc data, you will see new files generated. Note the time when the files were generated.

| Name | Last modified | Size | Storage class |
|--|-----------------------------------|---------|---------------|
| part-00000-00202004-2bc-47c-4249-d5100b75ddaa-c000.anappy.parquet | Jan 24, 2020 9:20:13 PM GMT-0500 | 9.3 kB | Standard |
| part-00000-49bea7fc-2ac1-4f87-e431-0e1f59d4a29-c000.anappy.parquet | Jan 24, 2020 7:03:16 PM GMT-0500 | 1.1 MB | Standard |
| part-00000-d1668723-315b-45b6-86d238402348-c000.anappy.parquet | Jan 25, 2020 11:24:20 PM GMT-0500 | 1.7 MB | Standard |
| part-00000-efcc0ff0-d440-43c-9320-c3d2a2119a-c000.anappy.parquet | Jan 25, 2020 10:24:27 PM GMT-0500 | 7.2 kB | Standard |
| part-00001-00202004-2bc-47c-4249-d5100b75ddaa-c000.anappy.parquet | Jan 24, 2020 9:20:13 PM GMT-0500 | 66.5 kB | Standard |
| part-00001-49bea7fc-2ac1-4f87-e431-0e1f59d4a29-c000.anappy.parquet | Jan 24, 2020 7:03:16 PM GMT-0500 | 1.2 MB | Standard |
| part-00001-d1668723-315b-45b6-86d238402348-c000.anappy.parquet | Jan 25, 2020 11:24:20 PM GMT-0500 | 1.7 MB | Standard |
| part-00002-00202004-2bc-47c-4249-d5100b75ddaa-c000.anappy.parquet | Jan 24, 2020 9:20:15 PM GMT-0500 | 1.7 MB | Standard |
| part-00002-d1668723-315b-45b6-86d238402348-c000.anappy.parquet | Jan 25, 2020 11:24:19 PM GMT-0500 | 1.5 MB | Standard |

2. Rerun the Glue job **Glue-Lab-TicketHistory-Parquet-with-bookmark** you created in Step 2
3. Go to the Athena Console, and rerun the following query to notice the increase in row count:

```
SELECT count(*) as recordcount FROM  
"ticketdata"."bookmark_parquet_ticket_purchase_history_data";
```

To review the latest transactions, run:

```
SELECT * FROM "ticketdata"."bookmark_parquet_ticket_purchase_history_data"  
order by transaction_date_time desc limit 100;
```

PART C: Glue Workflows (Optional, self-paced)

****Pre-requisite before creating workflow** - completed Part B**

Overview:

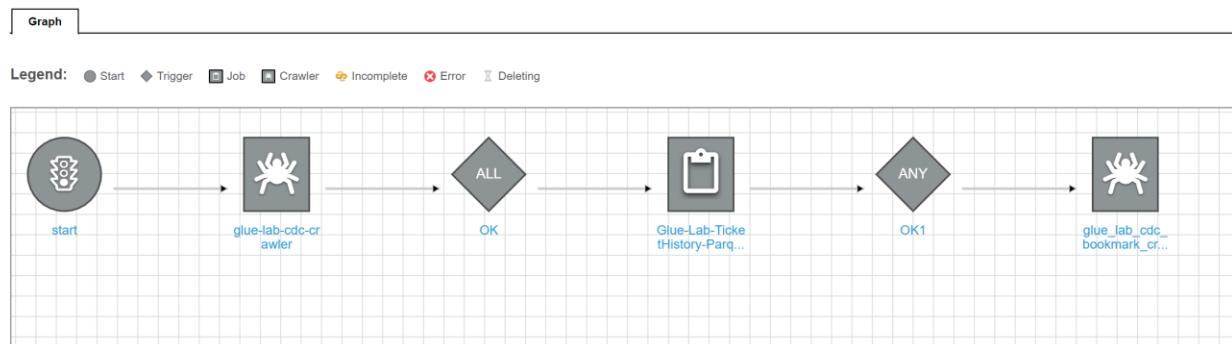
In AWS Glue, you can use workflows to create and visualize complex extract, transform, and load (ETL) activities involving multiple crawlers, jobs, and triggers. Each workflow manages the execution and monitoring of all its components. As a workflow runs each component, it records execution progress and status, providing you with an overview of the larger task and the details of each step. The AWS Glue console provides a visual representation of a workflow as a graph.

Creating and Running Workflows:

Above mentioned Part A (ETL with Glue) and Part B (Glue Job Bookmarks) can be created and executed using workflows. Complex ETL jobs involving multiple crawlers and jobs can also be created and executed using workflows in an automated fashion. Below is a simple example to demonstrate how to create and run workflows.

Try creating a new Glue Workflow to string together the two Crawlers and one Job from part B as follows:

On-demand trigger -> glue-lab-cdc-crawler -> Glue-Lab-TicketHistory-Parquet-with-bookmark -> glue_lab_cdc_bookmark_crawler



Congratulations!! You have successfully completed this lab